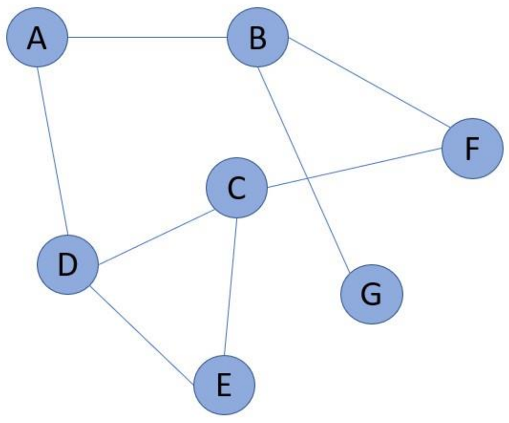
# Assignment: Graph Algorithms – I

*Note: The problem 2 is to be discussed as part of the Group Assignment. (Check this week's Group Assignment on Canvas for details).*

*The questions asked in this assignment – code implementation and time complexity of your code should be done individually based on the problem-solving strategy discussed within your group.*

1. **Write BFS and DFS for a graph**: What would be BFS and DFS traversal for the below graphs. Write the nodes for BFS and DFS. Start at node A.



**ANSWER:**

**BFS = A, B, D, F, G, C, E**

**DFS = A, B, F, C, D, E, G**

2. **Apply BFS/DFS to solve a problem**

You are given a 3-D puzzle. The length and breadth of the puzzle is given by a 2D matrix puzzle[m][n]. The height of each cell is given by the value of each cell, the value of puzzle[row][column] give the height of the cell [row][column]. You are at [0][0] cell and you want to reach to the bottom right cell [m-1][n-1], the destination cell. You can move either up, down, left, or right. Write an algorithm to reach the destination cell with minimal effort. How effort is defined: The effort of route is the maximum absolute difference between two consecutive cells.

If a route requires us to cross heights: 1, 3, 4, 6, 3, 1

The absolute differences between consecutive cells is: |1-3| = 2, |3-4|=1, |4-6|=2, |6-3|=3, |3-1|=2; this gives us the values: {2, 1, 2, 3, 2}. The maximum value of these absolute differences is 3. Hence the effort required on this path will be: 3.

Example:

Input: puzzle[][] = [[1, 3, 5], [2, 8, 3], [3, 4, 5]]

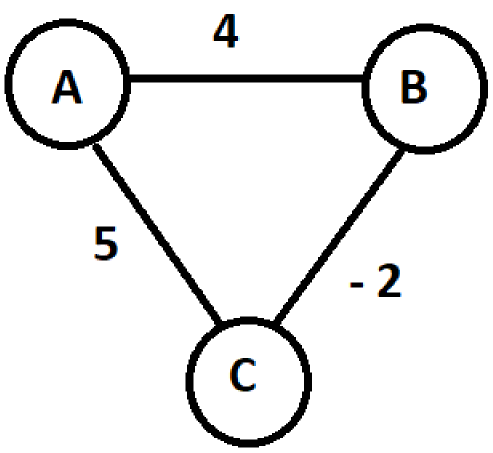
Output: 1
Explanation: The minimal effort route would be [1, 2, 3, 4, 5] which has an effort of value 1. This is better than other routes for instance, route [1, 3, 5, 3, 5] which has an effort of 2.

| | | |
|---|---|---|
| 1 | 3 | 5 |
| 2 | 8 | 3 |
| 3 | 4 | 5 |

a. Implement the algorithm. Name your function **minEffort(puzzle)**; puzzle will be in the form of an 2D matrix as shown in the above example. Name your file **MinPuzzle.py**
b. What is the time complexity of your implementation?

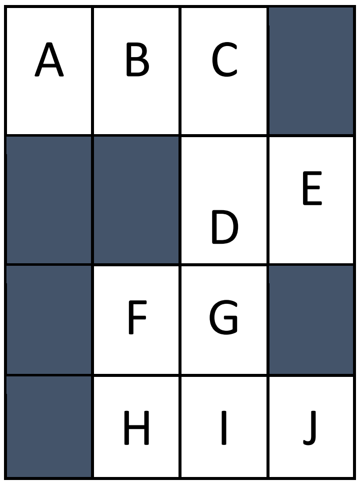
**ANSWER:** There are several steps in my code. First, building a dictionary with from a list of lists takes O(N) times. It uses two for loops but we are iterating through a list of lists which can be treated as a single list in time complexity, rather than O(N^2). The second part is using dijkstra's algorithm to find the minimal effort path. The only difference is we are storing the minimal effort rather than the sum of the distances. This has the time complexity of O((MN)logMN), M and N are the vertices and edges. Returning the value at [m-1][n-1] is O(1) when accessing the dictionary. We have a final complexity of O((MN)logMN) since that is a higher order of growth. The time complexity and PY solution was discussed and verified by TA Tim Pham.

**3. Analyze Dijkstra with negative edges: Analyze with a sample graph and show why Dijkstra does not work with negative edges. Give the sample graph and write your explanation why Dijkstra would not work in this case.**



Let's step through this to see why negative numbers don't work. Starting at node A, the cost is saved as 0 in the cost dictionary. Then nodes B and C are added to the priority queue. Node B is popped and cost 4 is added the cost dictionary. Node B is marked as visited. Node C is then popped and cost 2 (4 + -2) is saved to the cost dictionary. Node C is marked as visited. Now we have a cost of {A: 0, B: 4, C: 2}. Since A -> B -> C was considered as the shortest to C, edge A -> C is skipped. Node B could have been reached at a cost of 3 (5 + -2) if C was taken. Therefore, negative numbers don't work for Dijkstra's Algorithm.

**4. (Extra Credit): What would be BFS and DFS traversal in below puzzle. Start at node A.**



**DFS: ['A', 'B', 'C', 'D', 'E', 'G', 'F', 'H', 'I', 'J']  BFS: ['A', 'B', 'C', 'D', 'E', 'G', 'F', 'I', 'H', 'J']**

Debriefing (required!): -------------------------- Report:

Fill the report in the Qualtrics survey, you can access the link here.

(https://oregonstate.qualtrics.com/jfe/form/SV_agxCZlkLshpkvSS )

Note: ‘Debriefing’ section is intended to help us calibrate the assignments.